INITIALIZING/DIAGNOSING SYSTEM IN ON-CHIP MULTIPROCESSOR SYSTEM

BACKGROUND OF THE INVENTION

The present invention relates to an initializing/diagnosing system in an on-chip multiprocessor system, which is a multiprocessor system including a plurality of central processing elements (CPUs), and other elements which are diagnosed, such as a main memory element, a main memory access controller, and an input output processor (IOP) are mounted on a single Large Scale Integration (LSI).

A conventional on-chip multiprocessor system is shown in Fig. 6. An on-chip multiprocessor system 200 is connected to an initialization/diagnosis control processor 20. Initialization/diagnosis control processor 20 is a processor dedicated to initialization / diagnosis control implemented by another LSI. Initialization/diagnosis control processor 20 is independent of on-chip multiprocessor system 200. On-chip multiprocessor system 200 includes a CPU group 21 including CPUs 211 to 21m (m is an integer which satisfies m>0), a main memory access controller 22, a main memory element 23, and an IOP group 24 including IOPs 241 to 24n (n is an integer which satisfies n>0).

Initialization/diagnosis control processor 20 starts the operation of initializing/diagnosing CPU group 21, main memory access controller 22, main memory element 23, and IOP group 24 by receiving an electric confirmation signal, when power is applied to on-chip multiprocessor system 200, as a trigger signal from a system control section, which controls "system start-up" and "general corrective process" regarding on-chip multiprocessor system 200. In this embodiment, the electric confirmation signal is a CPU initialization/diagnosis start instruction signal outputted

from the system control section, and is typically represented by a signal notifying the application of power to on-chip multiprocessor system 200.

The initializing/diagnosing operation by initialization/diagnosis control processor 20 is performed by writing and reading data using a diagnosing path such as a scan path which is provided on each of the elements to be diagnosed, including CPU group 21, main memory access controller 22, main memory element 23, and IOP group 24.

The initialization/diagnosis control processor 20 outputs the diagnosed results to the system control section. The system control section disconnects the faulty elements , however, in some situations, some non-faulty elements may also be disconnected.

In the conventional system, an initialization/diagnosis control processor that is independent of the CPUs and non-CPU to-be-diagnosed elements must be provided. This creates a problem because the number of hardware elements required to drive the on-chip multiprocessor system is increased.

The initialization/diagnosis control processor is usually designed with an inexpensive technology to reduce the manufacturing cost of the on-chip multiprocessor system and thus, the clock frequency of the initialization/diagnosis control processor is lower than the clock frequency (clock frequency of the CPUs and the like) at normal operation of the on-chip multiprocessor system. This creates a problem because the initializing/diagnosing operation becomes slow,.

When the clock frequency of elements which are diagnosed, such as the CPUs, is different from the clock frequency of the initialization/diagnosis control processor, another problem occurs because it is necessary to also synchronize control.

A conventional diagnosing system for a multiprocessor is disclosed in Japanese laid-open publication Hei No. 3-19069. The diagnosing system for a multiprocessor includes a plurality of element processors, each of which compares its own processed

result with processed results of another element processors, and a majority circuit which receives the compared results from the processors and determines the presence of an abnormal element processor by using majority logic. The system has the majority circuit, however, it does not diagnose the entire on-chip multiprocessor system, including non-CPU elements.

SUMMARY OF THE INVENTION

An object of the invention is to provide an initializing/diagnosing system in an on-chip multiprocessor system that can eliminate the initialization/diagnosis control processor.

According to one aspect of the present invention, a semiconductor chip is provided which includes: a plurality of first elements, each of which diagnoses itself; and a second element which inputs diagnosis results from the first elements and determines whether any of the first elements are faulty.

According to another aspect of the present invention, a method which is performed in a semiconductor chip including a plurality of first elements is provided which includes: having the first elements diagnose themselves; and determining whether any of the first elements are faulty based on diagnosis results of the first elements.

BRIEF DESCRIPTION OF THE DRAWINGS

Other features and advantages of the invention will be made more apparent by the following detailed description and the accompanying drawings, wherein:

Fig. 1 is a block diagram of a first embodiment of the present invention;

Fig. 2 is a block diagram of a diagnosis circuit of the present invention;

Fig. 3 is a flowchart showing a process of the first embodiment of the present invention;

Fig. 4 is a block diagram of a second embodiment of the present invention;

Fig. 5 is a flowchart showing a process of the second embodiment of the present invention; and

Fig. 6 is a block diagram of a conventional on-chip multiprocessor system and initialization/diagnosis control processor.

In the drawings, the same reference numerals represent the same structural elements.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

A first embodiment of the present invention will be described in detail below.

An on-chip multiprocessor system 100 includes a CPU group 1, a main memory access controller 2, a main memory element 3, an IOP group 4, Read Only Memory (ROM) 5, and a diagnosis circuit 6. CPU group 1 includes CPUs 11 to 1m (m is an integer which satisfies m>0). IOP group 4 includes IOPs 41 to 4n (n is an integer which satisfies n>0). ROM 5 works as a program storing element. CPU group 1, main memory access controller 2, main memory element 3, IOP group 4, ROM 5, and diagnosis circuit 6 are mounted on a single LSI (the on-chip multiprocessor system).

ROM 5 stores an initializing/diagnosing program for CPU group 1 (CPU initializing/diagnosing program) and an initializing/diagnosing program for each of main memory access controller 2, main memory element 3, and IOP group 4 (hereinafter referred to as non-CPU to-be-diagnosed element initializing/diagnosing program). Elements such as main memory access controller 2, main memory element 3, and IOP group 4 are hereinafter referred to as non-CPU to-be-diagnosed elements.

Each of CPUs 11 to 1m loads the CPU initializing/diagnosing program from ROM 5 when starting own initializing/diagnosing operation. Each of CPUs 11 to 1m performs the initializing/diagnosing operation for the non-CPU to-be-diagnosed elements based on the non-CPU to-be-diagnosed element initializing/diagnosing programs when initializing/diagnosing operation for the non-CPU to-be-diagnosed elements. The initializing/diagnosing operation is similar to that in the prior art.

In this embodiment, the CPU initializing/diagnosing program and the non-CPU to-be-diagnosed element initializing/diagnosing programs can be individually stored in a plurality of program storing elements (ROMs).

Referring to Fig. 2, diagnosis circuit 6 includes a majority logic circuit group 60 having majority logic circuits 601 to 60x (x is an integer which satisfies x>0), an OR gate group 61 having OR gates 611 to 61m, and an encoder 62.

Majority logic circuits 601 to 60x are provided to correspond to pins of CPUs 11 to 1m which have the same pin configuration. Specifically, majority logic circuit 601 corresponds to a first output pin of each of CPUs 11 to 1m. Majority logic circuit 60x corresponds to an xth output pin of each of CPUs 11 to 1m. Each of majority logic circuits 601 to 60x outputs m outputs. Each of the m outputs correspond to each of CPUs 11 to 1m.

OR gates 611 to 61m are provided to correspond to CPUs 11 to 1m, respectively. Each of OR gates 611 to 61m inputs outputs, which correspond the same CPU, from majority logic circuits 601 to 60x. Specifically, OR gate 611 inputs outputs which correspond to CPU 11 from every majority logic circuit 601 to 60x. OR gate 61m inputs outputs which correspond to CPU 1m from every majority logic circuit 601 to 60x.

Referring to Fig. 3, a process of the initializing/diagnosing system in the on-chip multiprocessor system includes a trigger signal reception step 301, a CPU initializing/diagnosing program load step 302, a CPU initializing/diagnosing program execution start step 303, an output data notification step 304, a diagnosis circuit-made diagnosis and diagnosed result notification step 305, a system control section-addressed CPU group diagnosed result notification step 306, a non-CPU to-be-diagnosed element initializing/diagnosing program load step 307, a non-CPU to-be-diagnosed element group initializing/diagnosing operation execution step 308, and a system control section-addressed non-CPU to-be-diagnosed element group diagnosed result notification step 309.

Next, the operation of this embodiment will be described.

Upon application of power to the on-chip multiprocessor system, the on-chip multiprocessor system starts up, and CPUs 11 to 1m receive an electric confirmation signal as a trigger signal from a system control section (step 301).

When CPUs 11 to 1m receive the electric confirmation signal in step 301, each of CPUs 11 to 1m loads the CPU initializing/diagnosing program from ROM 5 to perform their own initializing/diagnosing operations (step 302), and starts executing the CPU initializing/diagnosing program (step 303).

Each of CPUs 11 to 1m outputs its output data, which is obtained during the execution of the CPU initializing/diagnosing program, to diagnosis circuit 6 (step 304). The output data from each of CPUs 11 to 1m is a set of output bits from x output pins.

Diagnosis circuit 6 receives the output data from each of CPUs 11 to 1m, diagnoses the presence or absence of a fault in each of CPUs 11 to 1m by a majority logic based on each output bit of the output data, and outputs an encoded value indicating the diagnosed result to each of CPUs 11 to 1m (step 305).

Next, the operation of diagnosis circuit 6 will be described in detail.

Referring to Fig. 2, majority logic circuits 601 receives m output bits from the first of the output pins (out pin 1) of CPUs 11 to 1m, respectively. Majority logic circuit 60x receives m output bits outputted from xth output pins (out pin x) of CPUs 11 to 1m, respectively. Majority logic circuits 601 outputs m outputs, each of which corresponds to each of m inputs (each of CPUs 11 to 1m), to each of OR gates 611 to 61m. Each of the majority logic circuits 601 to 60x obtains a majority logic of the m inputs (discriminates the m inputs into a majority group and a minority group), sends out "0" as outputs corresponding to the majority group inputs among the m inputs, and "1" as the outputs corresponding to the minority-group inputs.

That is, a majority logic circuit 60y (y is an integer which satisfies $1 < y < x$) receives m output bits outputted from every yth output pin of CPUs 11 to 1m, divides the received m inputs into a majority group and a minority group, and sets the output signals corresponding to the majority group inputs to "0" and sets the output signals corresponding to the minority group inputs to "1", and outputs the m output signals to OR gates 611 to 61m, respectively. For example, if none of CPUs 11 to 1m fail, all majority logic circuits 601 to 60x have m inputs (output bits of CPUs 11 to 1m) that are the same value, and output "0".

OR gates 611 receives outputs which corresponds to CPU 11 from each of majority logic circuits 601 to 60x. OR gate 61m receives outputs which corresponds to CPU 1m from each of majority logic circuits 601 to 60x. Each of OR gates 611 to 61m outputs an OR obtained from these received outputs. As a result, if any one of CPUs 11 to 1m fail, the OR gate corresponding to a faulty CPU outputs "1", which indicates that a CPU has failed.

Encoder 62 encodes the outputs of OR gates 611 to 61m and outputs the encoded values to all of the CPUs 11 to 1m. The encoded value contains information indicating the CPU number of a faulty CPU. If there are a plurality of faulty CPUs, then the CPU numbers of the faulty CPUs are reported to CPUs 11 to 1m over a predetermined period of time.

CPUs 11 to 1m recognize the presence or absence of a faulty CPU and a faulty CPU based on the encoded value from encoder 62. Because each of CPUs 11 to 1m holds its own CPU number in its register, each CPU can determine whether it is operating normally or not depending on whether the encoded value coincides with its own CPU number. The CPU number is set by configuration information held by the system control section at the time of start-up of the on-chip multiprocessor system.

Referring to Fig. 3, in step 306, a CPU in CPU group 1 having recognized that it is operating normally (hereinafter referred to as "normal CPU") outputs the diagnosed result of CPU group 1 to the system control section. For example, a normal CPU having the smallest CPU number in CPU group 1 notifies the system control section with information indicating the CPU number of a faulty CPU as the diagnosed result. The faulty CPU is degraded by the system control section at the end of the initializing/diagnosing operation for CPU group 1.

In step 307, when the initializing/diagnosing operation for CPU group 1 has completed, a group of normal CPUs in CPU group 1 loads, from ROM 5, the non-CPU to-be-diagnosed element initializing/diagnosing programs for main memory access controller 2, main memory element 3, and IOP group 4. The group of normal CPUs executes the initializing/diagnosing operation for the non-CPU to-be-diagnosed elements (step 308).

There are various ways to allot initializing/diagnosing operation to each normal CPU in the normal CPU group. It is likely that the operation is uniformly allotted to the normal CPUs and every normal CPUs performs shared pieces of initializing/diagnosing operation allotted to themselves simultaneously.

In step 309, when the initializing/diagnosing operation based on the non-CPU to-be-diagnosed element initializing/diagnosing programs has completed, the normal CPU group notifies the system control section of the diagnosed results of the non-CPU to-be-diagnosed element group. The elements determined to have a fault among the non-CPU to-be-diagnosed elements are disconnected, however, in some situations, some non-faulty elements may also be disconnected.

The configuration of diagnosis circuit 6 is not limited to that shown in Fig. 2, but may be of any type as long as the diagnosis circuit can diagnose the presence or absence of a fault in each of CPUs 11 to 1m in CPU group 1 using the majority logic.

Further, the elements belonging to the non-CPU to-be-diagnosed element group are not limited to those shown in Fig. 1, such as, main memory access controller 2, main memory element 3, and IOP group 4.

As described above, it is possible to reduce the number of hardware elements required to drive the on-chip multiprocessor system because a processor dedicated to initialization/diagnosis control (initialization/diagnosis control processor) can be eliminated.

The initializing/diagnosing operation can be performed at a clock frequency during normal operation of the on-chip multiprocessor system because the LSI (the on-chip multiprocessor system), makes a self-diagnosis. Therefore, the present invention can perform the initializing/diagnosing operation at higher speeds than the prior art that uses the initialization/diagnosis control processor.

Because the elements subjected to the initializing/diagnosing operation are mounted on the LSI (the on-chip multiprocessor system), synchronous control of clock frequency between the to-be-diagnosed elements including CPUs and the initialization/diagnosis control processor is not required.

It is also possible to reduce the manufacturing cost of the LSI implementing the on-chip multiprocessor system because the LSI implementing the on-chip multiprocessor itself has a sophisticated self-diagnosis function as described above and thus an inspection step based on the function of an LSI tester can be omitted.

Next, a second embodiment of the present invention will be described in detail.

A feature of the second embodiment is configuration which achieve system evaluation and efficiency in failure analysis.

Referring to Fig. 4, an on-chip multiprocessor system 110 includes a CPU group 1 consisting of CPUs 11 to 1m, a main memory access controller 2, a main memory element 3, an IOP group 4 consisting of IOPs 41 to 4n, a ROM 5 which works as a program storing element, a diagnosis circuit 6, a load path selection circuit 7, a trigger signal selection circuit 8, and a mode register 9 that implements a mode information storing element which stores mode information that consists of two bits, i.e., the 0th bit and the first bit. CPU group 1, main memory access controller 2, main memory element 3, IOP group 4, ROM 5, diagnosis circuit 6, load path selection circuit 7, trigger signal selection circuit 8, and mode register 9 are mounted on a single LSI (the on-chip multiprocessor system).

In Fig. 5, the process of initializing/diagnosing the system in the on-chip multiprocessor system includes a load path mode determination step 501, a via-ROM-data-input load path selection step 502, a via-external-data-input load path selection step 503, a trigger signal mode determination step 504, an electric confirmation signal

(as a trigger signal) selection step 505, a user set signal (as a trigger signal) selection step 506, a trigger signal reception step 507, a CPU initializing/diagnosing program load step 508, a CPU initializing/diagnosing program execution start step 509, an output data notification step 510, a diagnosis circuit-made diagnosis and diagnosed result report step 511, a system control section-addressed CPU group diagnosed result notification step 512, a non-CPU to-be-diagnosed element initializing/diagnosing program load step 513, a non-CPU to-be-diagnosed element group initializing/diagnosing operation execution step 514, and a system control section-addressed non-CPU to-be-diagnosed element group diagnosed result notification step 515.

The detailed configuration of diagnosis circuit 6 is shown in Fig. 2.

Next, the operation of this embodiment will be described.

The mode register 9 holds mode information in the 0th and first bits as set by the user in advance. The following items 1) and 2) show the mode information:

1) When a path via a data input from ROM 5 is selected as an initializing/diagnosing program load path, the 0th bit is set to "0". On the other hand, when a load path via a data input from an external input is selected, the 0th bit is set to "1".

2) When a trigger signal that triggers the start of the initializing/diagnosing operation by CPUs 11 to 1m is used as "an electric confirmation signal outputted from the system control section when the power is applied to the on-chip multiprocessor," the first bit is set to "0". On the other hand, when the trigger signal is used as "a user set signal that the user specified (entered) from a console, or the like, when the user wishes to trigger the operation himself," the first bit is set to "1".

The aforementioned settings are merely one example.

In step 501, load path selection circuit 7 determines whether the 0th bit of mode register 9 is set to "0" or "1". If the bit is set to "0", load path selection circuit 7 selects to load the initializing/diagnosing programs (the CPU initializing/diagnosing program and the non-CPU to-be-diagnosed element initializing/diagnosing programs) via a data input from ROM 5 (step 502). If the 0th bit of mode register 9 is set to "1", the circuit 7 loads the initializing/diagnosing programs via a data input from an external input (step 503). In this way, load path selection circuit 7 switches the paths through which CPUs 11 to 1m load the initializing/diagnosing programs.

In step 504, trigger signal selection circuit 8 determines whether the first bit of mode register 9 is set to "0" or "1". If the bit is set to "0", the circuit 8 uses the electric confirmation signal as a trigger signal that triggers the start of the initializing/diagnosing operation (step 505). If the first bit of mode register 9 is set to "1", trigger signal selection circuit 8 uses the user set signal as the trigger signal (step 506). In this way, trigger signal selection circuit 8 implements trigger signal switching.

Upon reception of the trigger signal (the electric confirmation signal or user set signal) selected by trigger signal selection circuit 8 in step 507, CPUs 11 to 1m load the CPU initializing/diagnosing program recorded in ROM 5 or the CPU initializing/diagnosing program based on a data input from an external input via the load path set based on the selection made by load path selection circuit 7 (step 508). In step 509, CPUs 11 to 1m start executing the loaded CPU initializing/diagnosing program.

In steps 510 to 512, the initializing/diagnosing operation for CPUs 11 to 1m is performed similarly to the operation in the on-chip multiprocessor according to the first embodiment. It is noted that steps 510 to 512 in Fig. 5 correspond to steps 304 to 306 in Fig. 3.

Further, at the end of the initializing/diagnosing operation for CPU group 1, a group of normal CPUs in CPU group 1 load the non-CPU to-be-diagnosed element initializing/diagnosing programs for the operation of initializing/diagnosing main memory access controller 2, main memory element 3, and IOP group 4 through the load path which is set based on the selection made by load path selection circuit 7. Specifically, the normal CPUs load the non-CPU to-be-diagnosed element initializing/diagnosing programs recorded in ROM 5 or the non-CPU to-be-diagnosed element initializing/diagnosing programs based on a data input from an external input (step 513), and perform the initializing/diagnosing operation for the aforementioned non-CPU to-be-diagnosed elements based on the loaded non-CPU to-be-diagnosed element initializing/diagnosing programs (step 514).

In step 515, the initializing/diagnosing operation for the non-CPU to-be-diagnosed elements is performed similarly to the operation in the on-chip multiprocessor according to the first embodiment. It is noted that step 515 in Fig. 5 corresponds to step 309 in Fig. 3.

As described above, this embodiment allows the user to variably set the timing for triggering the start of the initializing/diagnosing operation. Thus, even if the user finds the problems encountered at the time of the on-chip multiprocessor system start-up, this embodiment is advantageous because system evaluation can be performed efficiently and effectively. A problem encountered at the time of the on-chip multiprocessor system start-up, for example, is created because the problems that the on-chip multiprocessor system cannot be started up normally when logic bugs and failures are found in the CPUs and non-CPU to-be-diagnosed elements, and when there are bugs in initially set data, and the like.

Further, this embodiment allows the initializing/diagnosing programs to be executed by not only using the initializing/diagnosing programs recorded in ROM 5, but by also using the initializing/diagnosing programs based on data input from an external input. Therefore, various initializing/diagnosing programs can be executed. Hence, this embodiment is advantageous because system evaluation and failure analysis can be performed more efficiently and effectively.

It is noted that mode switching can be effected only for one mode selected from the initializing/diagnosing program load path-related mode or the trigger signal-related mode. A configuration to be adopted in such a case is shown in items 1) and 2) below:

1) When switching is made only for the initializing/diagnosing program load path-related mode, the configuration is such that load path selection circuit 7 and mode register 9, which holds only the 0th bit as the mode information, both shown in Fig. 4, are added to the basic configuration shown in Fig. 1.

2) When switching is made only for the trigger signal-related mode, the configuration is such that trigger signal selection circuit 8 and mode register 9, which holds only the first bit as the mode information, both shown in Fig. 4, are added to the basic configuration shown in Fig. 1.

The configuration of diagnosis circuit 6 is not limited to that shown in Fig. 2, but may be of any type as long as circuit 6 can diagnose the presence or absence of a fault in each of CPUs 11 to 1m in CPU group 1 using the majority logic as in the first embodiment. In addition, the non-CPU to-be-diagnosed elements are not limited to those shown in Fig. 4 (main memory access controller 2, main memory element 3, and IOP group 4) as in the first embodiment.

While this invention has been described in conjunction with the preferred embodiments described above, it will now be possible for those skilled in the art to put this invention into practice in various other manners.